I CLAIM:

1. Apparatus for processing data, said apparatus comprising:

(i) a processor core;

(ii) a main memory operable to store instruction words and data words;

(iii) a data store operable to store words from said main memory accessed by a data store port of said processor core;

(iv) an instruction store operable to store words from said main memory accessed by an instruction store port of said processor core; and

(v) an instruction interpreter operable to read instruction words from said instruction store; wherein

(vi) said instruction interpreter is operable to modify a slow form instruction within said instruction store to a fast form instruction of one or more possible fast form instructions and to write said fast form instruction to said data store, said slow form instruction and said fast form instruction having a common functionality when executed by said interpreter; and

(vii) said instruction interpreter is operable upon reading a slow form instruction from said instruction store to check for a corresponding fast form instruction within said data store and, if said fast form instruction is present within said data store, then to execute said fast form instruction instead of said slow form instruction.

2. Apparatus as claimed in claim 1, wherein said instruction interpreter is a hardware based instruction translator.

3. Apparatus as claimed in claim 1, wherein said instruction interpreter is a software based interpreter.

4. Apparatus as claimed in claim 1, wherein said instruction interpreter is a combination of a hardware based instruction translator and a software based interpreter.

5. Apparatus as claimed in claim 1, wherein said data store is a data cache and said data store port is a data cache port.

6. Apparatus as claimed in claim 1, wherein said instruction store is an instruction cache and said instruction store port is an instruction cache port.

7. Apparatus as claimed in claim 1, wherein said slow form instruction results in an unresolved storage access request to one or more stored words and said fast form instruction results in a resolved storage access request to said one or more stored words.

8. Apparatus as claimed in claim 1, wherein said slow form instruction includes a symbolic reference to a required element and said fast form instruction includes a numeric reference to said required element.

9. Apparatus as claimed in claim 1, wherein said slow form instruction invokes an additional data processing procedure before completion.

10. Apparatus as claimed in claim 1, wherein said slow form instruction and said fast form instruction are Java Virtual Machine instructions.

11. Apparatus as claimed in claim 10, wherein said slow form instruction is one of:

anewarray;
checkcast;
getfield;
getstatic;
instanceof;
invokeinterface;
invokespecial;
invokestatic;
invokevirtual;
ldc;
ldc_w;
ldc2_w;
multianewarray;
new;
putfield; and
putstatic.

12. Apparatus as claimed in claim 10, wherein said fast form instruction is one of:

anewarray_quick;

checkcast_quick;

getfield_quick;

getfield_quick_w;

getfield2_quick;

getstatic_quick;

getstatic2_quick;

instanceof_quick;

invokeinterface_quick;

invokenonvirtual_quick;

invokesuper_quick;

invokestatic_quick;

invokevirtual_quick;

invokevirtual_quick_w;

invokevirtualobject_quick;

ldc_quick;

ldc_w_quick;

ldc2_w_quick;

multianewarray_quick;

new_quick;

putfield_quick;

putfield_quick_w;

putfield2_quick;

putstatic_quick; and

putstatic2_quick.

13. Apparatus as claimed in claim 10, wherein said instruction interpreter translates Java Virtual Machine instructions to native instructions of said processor core.

14. A method of processing data using an apparatus having a processor core, a main memory operable to store instruction words and data words, a data store operable to store words from said main memory accessed by a data store port of said processor core, an

instruction store operable to store words from said main memory accessed by an instruction store port of said processor core, and an instruction interpreter operable to read instruction words from said instruction store; said method comprising the steps of:

(i) modifying a slow form instruction within said instruction store to a fast form instruction of one or more possible fast form instructions and to write said fast form instruction to said data store, said slow form instruction and said fast form instruction having a common functionality when executed by said interpreter; and

(ii) upon reading a slow form instruction from said instruction store, checking for a corresponding fast form instruction within said data store and, if said fast form instruction is present within said data store, then executing said fast form instruction instead of said slow form instruction.